DBMS MINI PROJECT

RESORT MANAGEENT SYSTEM : -

Submitted By :

SATISH G HUDDAR

PES1UG20CS590

V Semester J-Section

# Project Description :

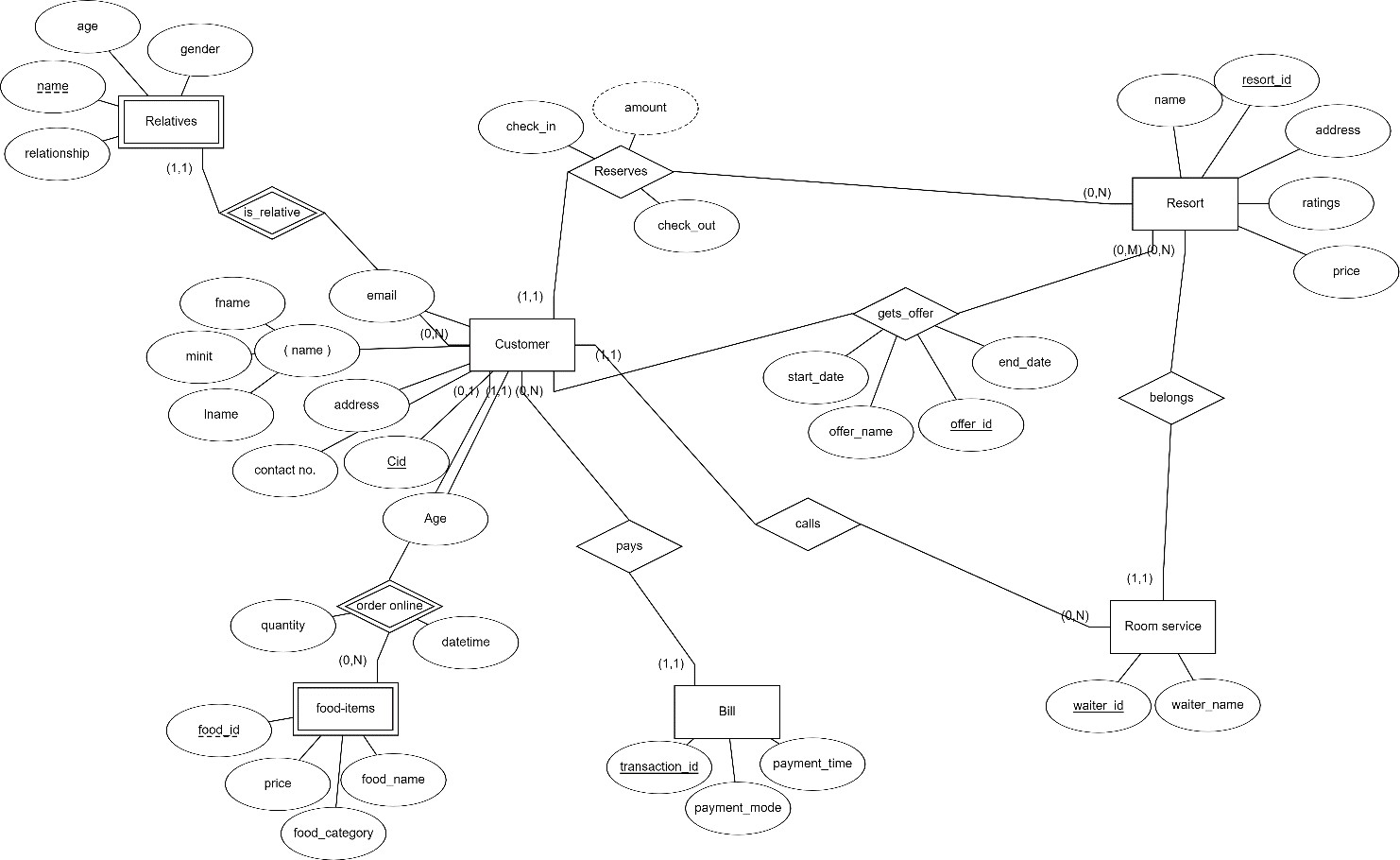
Traditionally, all the information of the customers who have booked a particular resort is stored in books/ledgers. This is very inefﬁcient and labour intensive. This problem can be solved using SQL database, which is stored digitally and information is safe.

The database consists of several entities like customer, resort, offers, food\_item, reservation, bill,. Etc which contain all the necessary information.

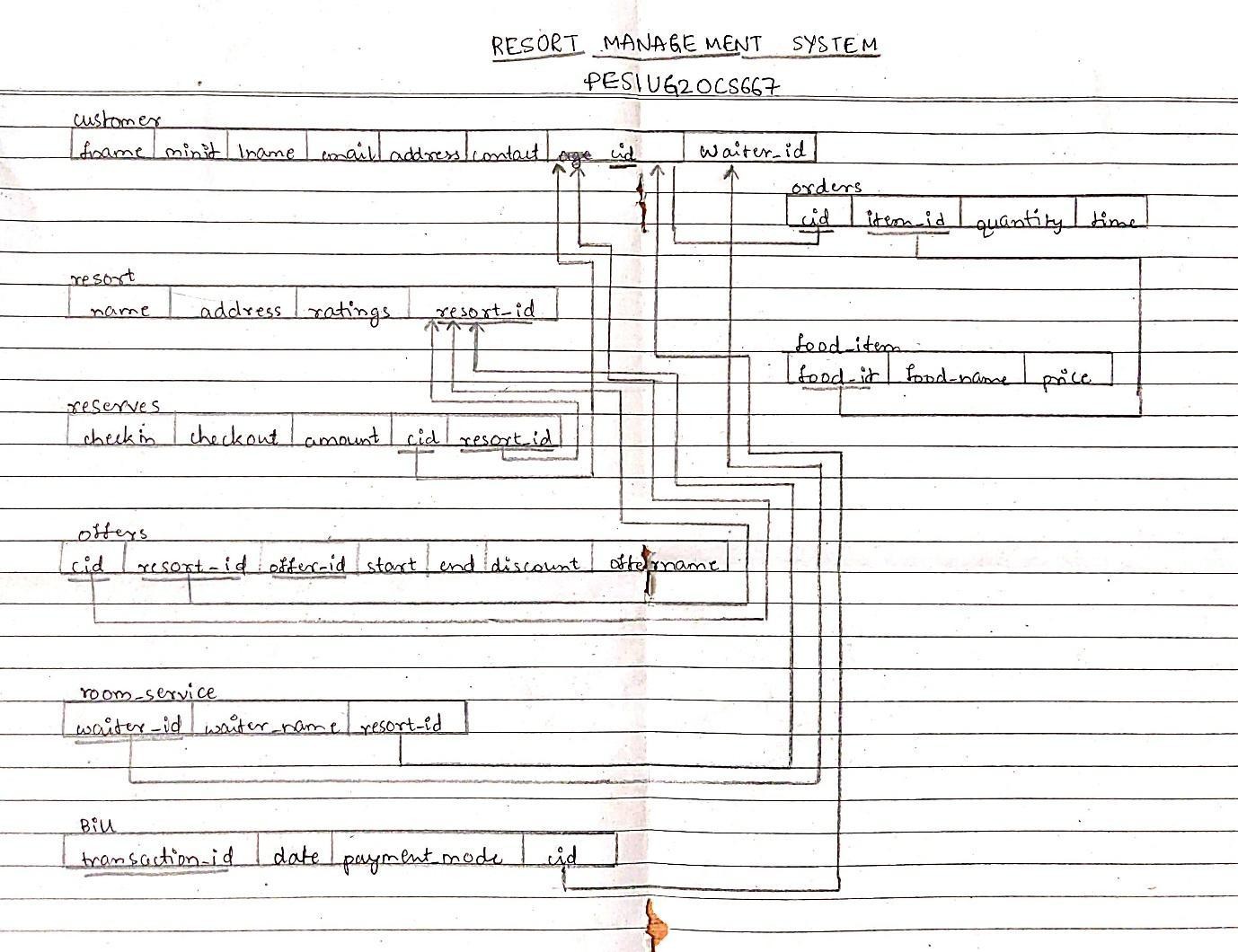
# Scope :

To create a website system for managing resort reservations and make it easier for the user to book the resort and buy ticket

# Entity Relationship (ER) Diagram :



**Relational Schema :**



DDL Statements-Building the Database

CREATE TABLE resort (

resort\_id DECIMAL(4, 0) PRIMARY KEY CHECK (resort\_id > 0), resort\_name varchar(50) NOT NULL,

address varchar(40) NOT NULL, rating DECIMAL(3,2), price\_per\_day float

);

CREATE TABLE room\_service (

waiter\_id DECIMAL(3, 0) PRIMARY KEY,

waiter\_name varchar(20), resort\_id DECIMAL(4, 0),

FOREIGN KEY (resort\_id) REFERENCES resort(resort\_id) ON DELETE CASCADE

);

CREATE TABLE customer (

cid Decimal(4, 0) PRIMARY KEY CHECK (cid > 0), fname VARCHAR(20),

minit CHAR(1), lname VARCHAR(20),

address varchar(30), email varchar(30), contactNo DECIMAL(10, 0),

waiter\_id DECIMAL(3, 0),

FOREIGN KEY (waiter\_id) REFERENCES room\_service(waiter\_id) ON DELETE CASCADE

);

CREATE TABLE offers ( offer\_id DECIMAL(2, 0), offer\_name varchar(20),

cid DECIMAL(4, 0) CHECK (cid > 0),

resort\_id DECIMAL(4, 0) CHECK (resort\_id > 0), discount int,

startdate DATE, enddate DATE,

FOREIGN KEY (resort\_id) REFERENCES resort(resort\_id) ON DELETE CASCADE, FOREIGN KEY (cid) REFERENCES customer(cid) ON DELETE CASCADE,

PRIMARY KEY(resort\_id, cid, offer\_id)

);

CREATE TABLE reservation(

cid DECIMAL(4, 0) CHECK (cid > 0),

resort\_id DECIMAL(4, 0) CHECK (resort\_id > 0), checkin DATE NOT NULL,

checkout DATE NOT NULL, amount float,

CHECK (checkout > checkin),

FOREIGN KEY (cid) REFERENCES customer(cid) ON DELETE CASCADE,

FOREIGN KEY (resort\_id) REFERENCES resort(resort\_id) ON DELETE CASCADE, PRIMARY KEY(resort\_id, cid)

);

CREATE TABLE relatives( cid DECIMAL(4, 0),

relative\_name VARCHAR(20), gender char(1), relationship VARCHAR(20),

FOREIGN KEY (cid) REFERENCES customer(cid) ON DELETE CASCADE ON UPDATE CASCADE,

PRIMARY KEY (cid, relative\_name)

);

CREATE TABLE food\_item ( food\_id decimal(2, 0), food\_name varchar(20), price numeric CHECK (

price BETWEEN 0.00 AND 500.00

),

PRIMARY KEY(food\_id)

);

= customer orders food via some food delivery app CREATE TABLE orders (

cid DECIMAL(4, 0),

item\_id DECIMAL(3, 0), quantity INT,

time DATETIME,

FOREIGN KEY(cid) REFERENCES customer(cid) ON DELETE RESTRICT,

FOREIGN KEY(item\_id) REFERENCES food\_item(food\_id) ON DELETE RESTRICT, PRIMARY KEY (cid, item\_id)

);

= number of digits in transaction Id varies, 12 is most common CREATE TABLE bill (

transaction\_id DECIMAL(12, 0) PRIMARY KEY, date DATE,

cid DECIMAL(4, 0),

paymentmode varchar(10),

FOREIGN KEY(cid) REFERENCES customer(cid) ON DELETE RESTRICT

);

DML Statements-Populating the Database

insert into food\_item Values('01','thaali','400');

insert into food\_item Values('02','pizza','250');

insert into food\_item Values('03','ghee rice','120');

insert into food\_item Values('04','schezwan fried rice','120');

insert into food\_item Values('05','chicken biryani','180');

insert into food\_item Values('06','mutton thaali','320');

insert into food\_item Values('07','surma fish','240');

insert into offers Values('01','winter vaction', '1001','1006','20','2022-12-20','2022-12-30');

insert into offers Values('02','diwali offer', '1005','1001','15','2022-10-01','2022-12-31');

insert into offers Values('03','special offer', '1007','1007','25','2022-11-01','2022-12-31');

insert into resort Values ('1001',"The Dukes Retreat",'Lonavala',5,1999);

insert into resort Values ('1002',"Ferreira Resort",'Lonavala',4,1799);

insert into resort Values ('1003',"Villa San Lorentz",'Lonavala',5,1699);

insert into resort Values ('1004',"Misty Meadows",'Lonavala',4,1499);

insert into resort Values ('1005',"Sunshine Resort",'Lonavala',5,1699);

insert into resort Values ('1006',"Dandeli Jungle Resort",'Dandeli',5,1499);

insert into resort Values ('1007',"Wild Planet Jungle Resort",'Dandeli',5,1299);

insert into resort Values ('1008',"Swast-Mast Resort",'Lonavala',4,1399);

insert into resort Values ('1009',"Alurkar Resort",'Belgaum',5,1499);

insert into resort Values ('1010',"Gavkari",'Belgaum',5,999);

insert into room\_service Values('101','chotu','1001');

insert into room\_service Values('102','bhola','1002');

insert into room\_service Values('103','brijesh','1003');

insert into room\_service Values('104','ajay','1004');

insert into room\_service Values('105','shukh','1005');

insert into room\_service Values('106','vikalp','1006');

insert into room\_service Values('107','alam','1007');

insert into room\_service Values('108','suresh','1008');

insert into room\_service Values('109','keshav','1006');

insert into room\_service Values('110','sharad','1009');

insert into room\_service Values('111','munna','1010');

insert into customer Values('1001','narendra','','modi','gujarat' ,'modi@gmail.com' ,'1234567890','106');

insert into customer Values('1002','amit','','shah','gujarat' ,'shah@gmail.com' ,'1234567890','102');

insert into customer Values('1003','atal','b','vajpayee','bihar' ,'vajpayee@gmail.com' ,'1234567890','103');

insert into customer Values('1004','abdul','','kalam','patna' ,'aniket@gmail.com' ,'1234567890','104');

insert into customer Values('1005','yogi','','adityanath','uttar pradesh' ,'yogi@gmail.com' ,'1234567890','102');

insert into customer Values('1006','balasaheb','','thakre','mumbai' ,'thakre@gmail.com' ,'1234567890','101');

insert into customer Values('1007','basavraj','','bommai','karnataka' ,'bommai@gmail.com' ,'1234567890','102');

insert into bill Values('250707244234','2022-11-21','1001','upi');

insert into bill Values('202622637838','2022-11-22','1002','debit card');

insert into bill Values('644161913172','2022-11-15','1003','credit card');

insert into bill Values('988602103725','2022-11-21','1004','upi');

insert into bill Values('859741505883','2022-11-21','1005','cash');

insert into bill Values('346488919858','2022-11-26','1006','credit card');

insert into orders Values('1001','1','3','2022-11-21  10:55:05');

insert into orders Values('1002','4','4','2022-11-21  13:56:55');

insert into orders Values('1003','5','2','2022-11-10  12:35:05');

insert into orders Values('1004','7','8','2022-11-20  21:55:05');

insert into orders Values('1002','3','1','2022-11-18  07:24:05');

insert into orders Values('1006','4','4','2022-11-25  09:18:05');

insert into relatives Values('1002','bhaskar bhat','m','friend');

insert into relatives Values('1003','vijay verma','m','friend');

insert into relatives Values('1003','sujay patil','m','friend');

insert into relatives Values('1004','sharad shukla','m','friend');

insert into relatives Values('1004','anil desai','m','son');

insert into relatives Values('1005','manish gupta','m','colleague');

insert into relatives Values('1005','kartik singh','m','colleague');

insert into relatives Values('1005','shilpa trivedi','f','colleague');

insert into relatives Values('1006','manthan patil','m','friend');

insert into reservation values('1001','1001','2022-11-11','2022-11-21',15592);

insert into reservation values('1002','1002','2022-11-15','2022-11-22',12593);

insert into reservation values('1003','1003','2022-11-15','2022-11-227',11893);

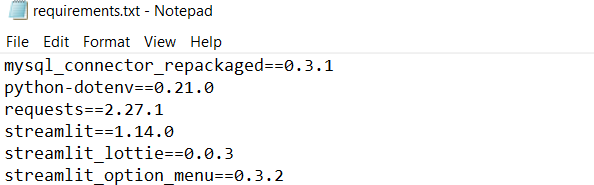
insert into reservation values('1004','1004','2022-11-16','2022-11-21',7495);

insert into reservation values('1005','1005','2022-11-18','2022-11-21',4077.6);

insert into reservation values('1006','1006','2022-11-20','2022-11-26',8994);

## Tools Used

* UI for database operations – streamlit
* Database connection – mysql-connector-python
* Xampp

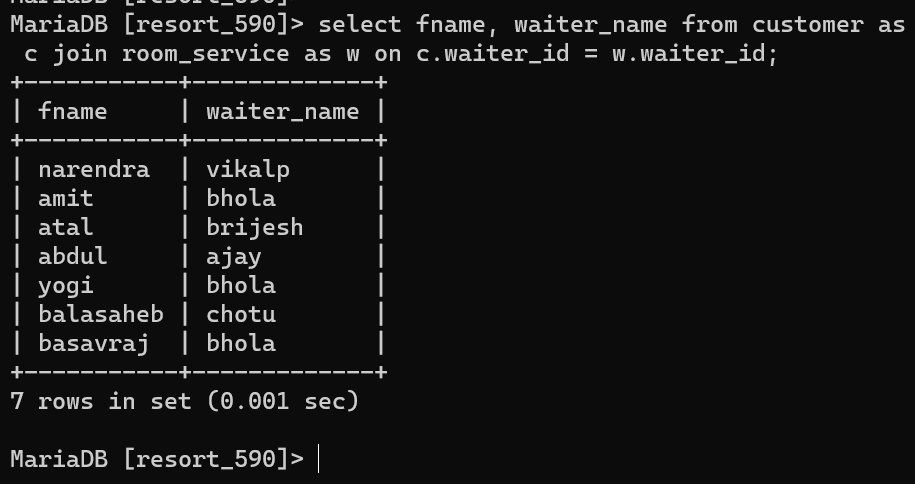


## QUERIES

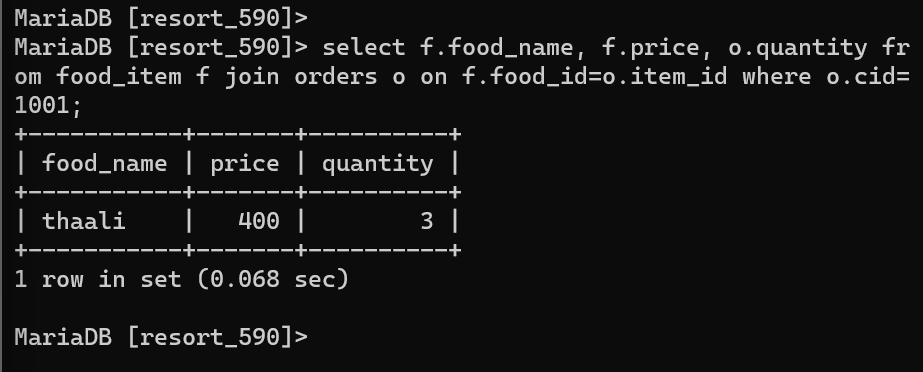
JOIN QUERIES

1. **Display ﬁrst name of customer and name waiter assigned to him**

select fname, waiter\_name from customer as c join room\_service as w on c.waiter\_id = w.waiter\_id;



1. **Display food\_name, price and quantity of all food orders made by customer with cid=1001**

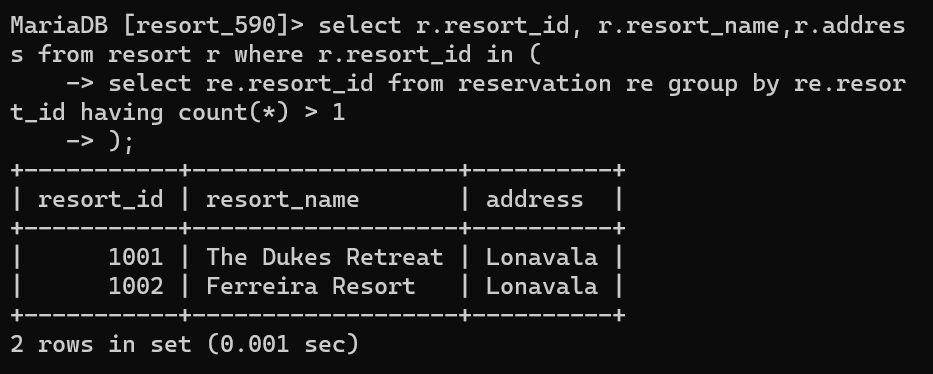


1. **Display resort\_id, resort name and address of all those resorts where the number of bookings is greater than 1.**

select r.resort\_id, r.resort\_name,r.address from resort r where r.resort\_id in (

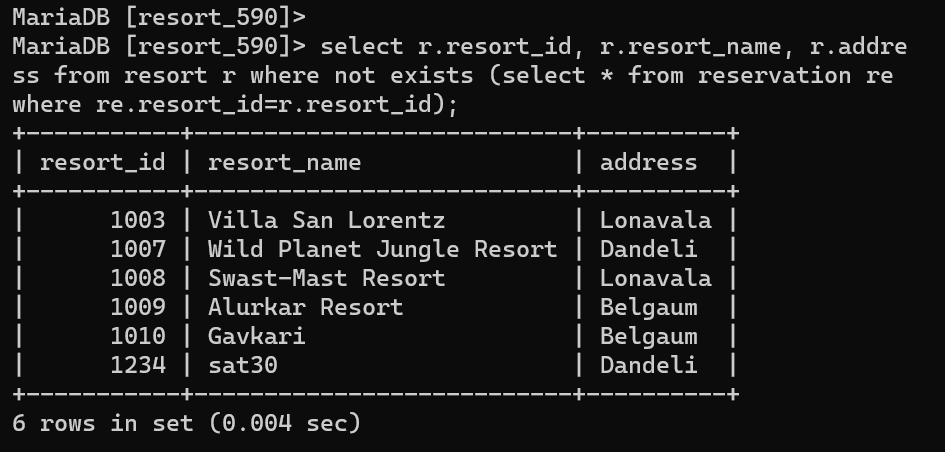
select re.resort\_id from reservation re group by re.resort\_id having count(\*) > 1

);



1. **Display information of resort which are available for booking**

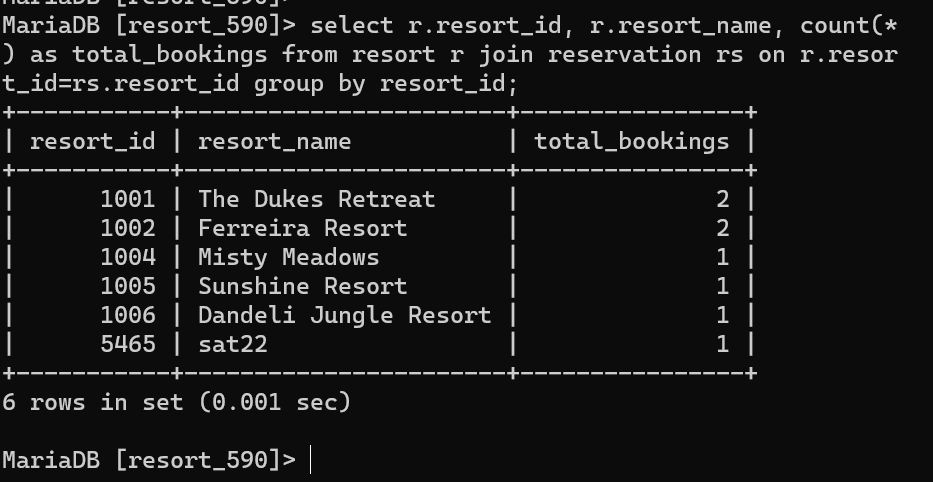
select r.resort\_id, r.resort\_name, r.address from resort r where not exists (select \* from reservation re where re.resort\_id=r.resort\_id);



## AGGREGATE FUNCTIONS

1. **Display total bookings for each resort**

select r.resort\_id, r.resort\_name, count(\*) as total\_bookings from resort r join reservation rs on r.resort\_id=rs.resort\_id group by resort\_id;



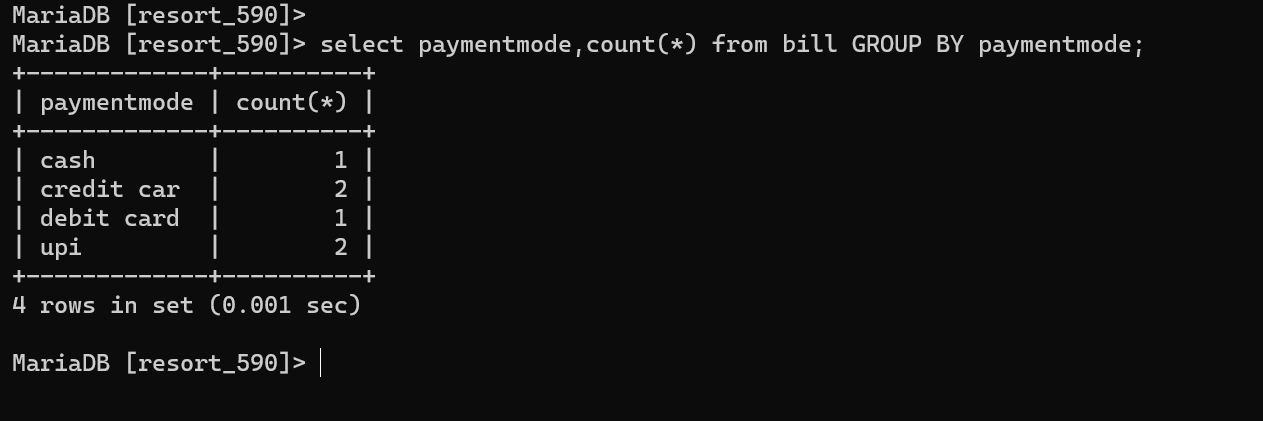
1. **List all resorts city wise**

select r.address as location, count(\*) total\_resorts from resort r group by r.address;



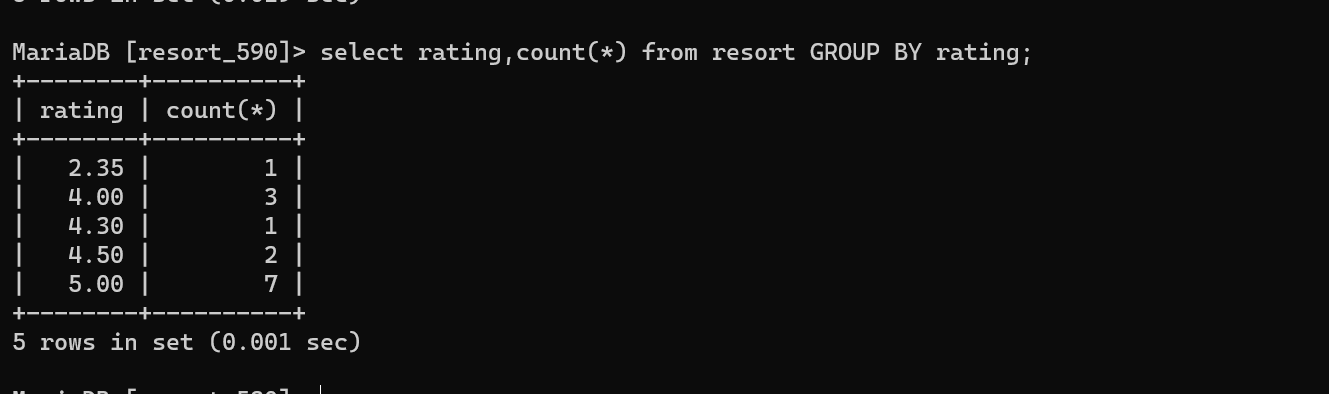
1. **To find number of payments done through different modes**

select paymentmode,count(\*) from bill GROUP BY paymentmode;

****

1. **Number of resorts Rating wise**

select rating,count(\*) from resort GROUP BY rating;

****

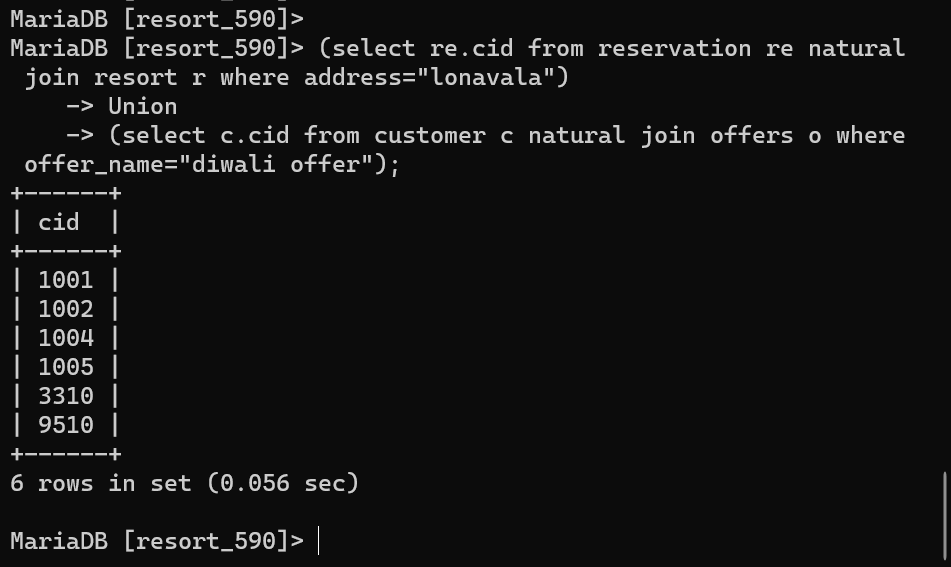
## SET OPERATIONS

1. **List all customers who have booked a resort with “Diwali offer” in Lonavala.**

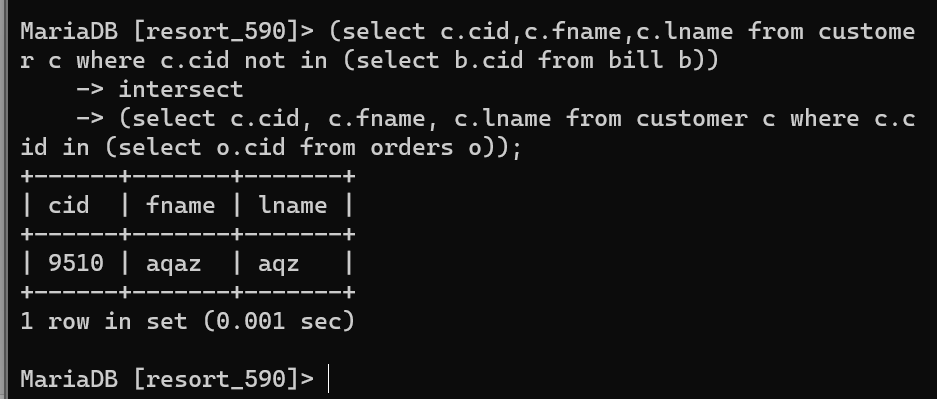
(select re.cid from reservation re natural join resort r where address="lonavala")

Union

(select c.cid from customer c natural join offers o where offer\_name="diwali offer");

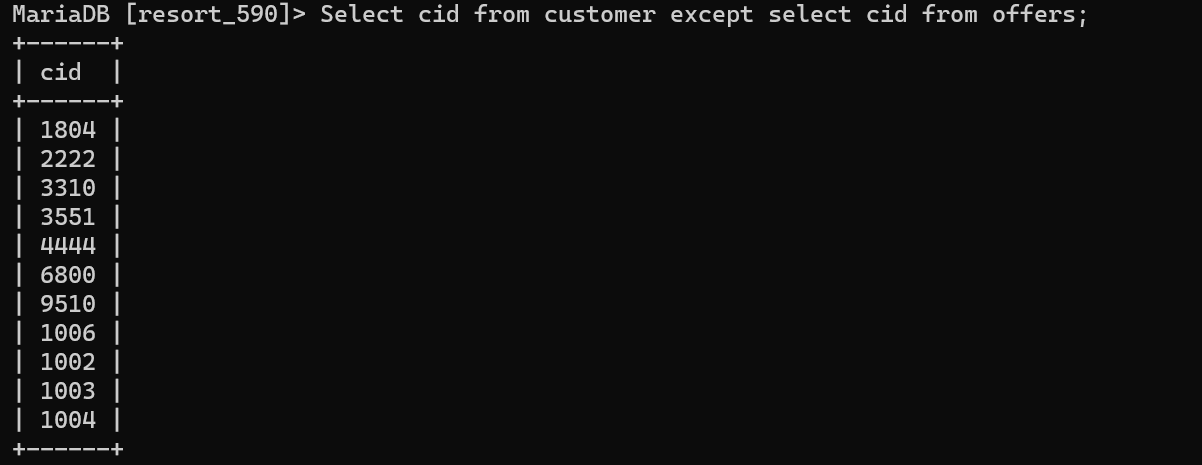


1. **List all customers who have NOT paid the bill but have ordered food item**



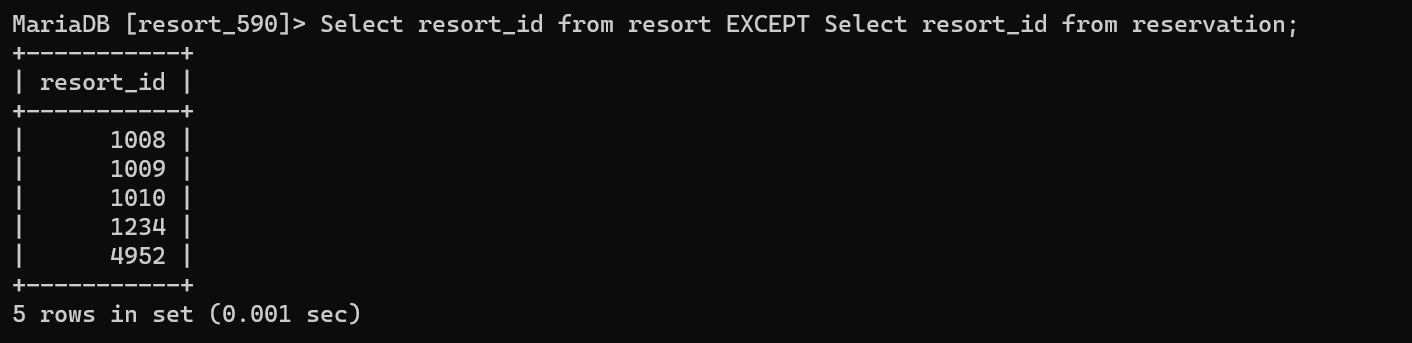
1. **List Customer who not got any offer**

Select cid,fname from customer except select cid from offers;



1. **To find resort which not have reserved**

Select resort\_id from resort EXCEPT Select resort\_id from reservation;



## FUNCTIONS

1. To convert rating to feedback
2. DELIMITER $$
3. CREATE FUNCTION ratef(rate decimal(1))
4. RETURNS VARCHAR(50)
5. DETERMINISTIC
6. BEGIN
7. IF rate < 3 THEN
8. RETURN "Average";
9. ELSE
10. RETURN "Very Good";
11. END IF;
12. END$$
13. DELIMITER ;



## STORED PROCEDURE

1. display information of all resorts in a given city

DELIMITER $$

CREATE OR REPLACE PROCEDURE get\_\_resort\_count(IN city varchar(15), OUT r\_count integer)

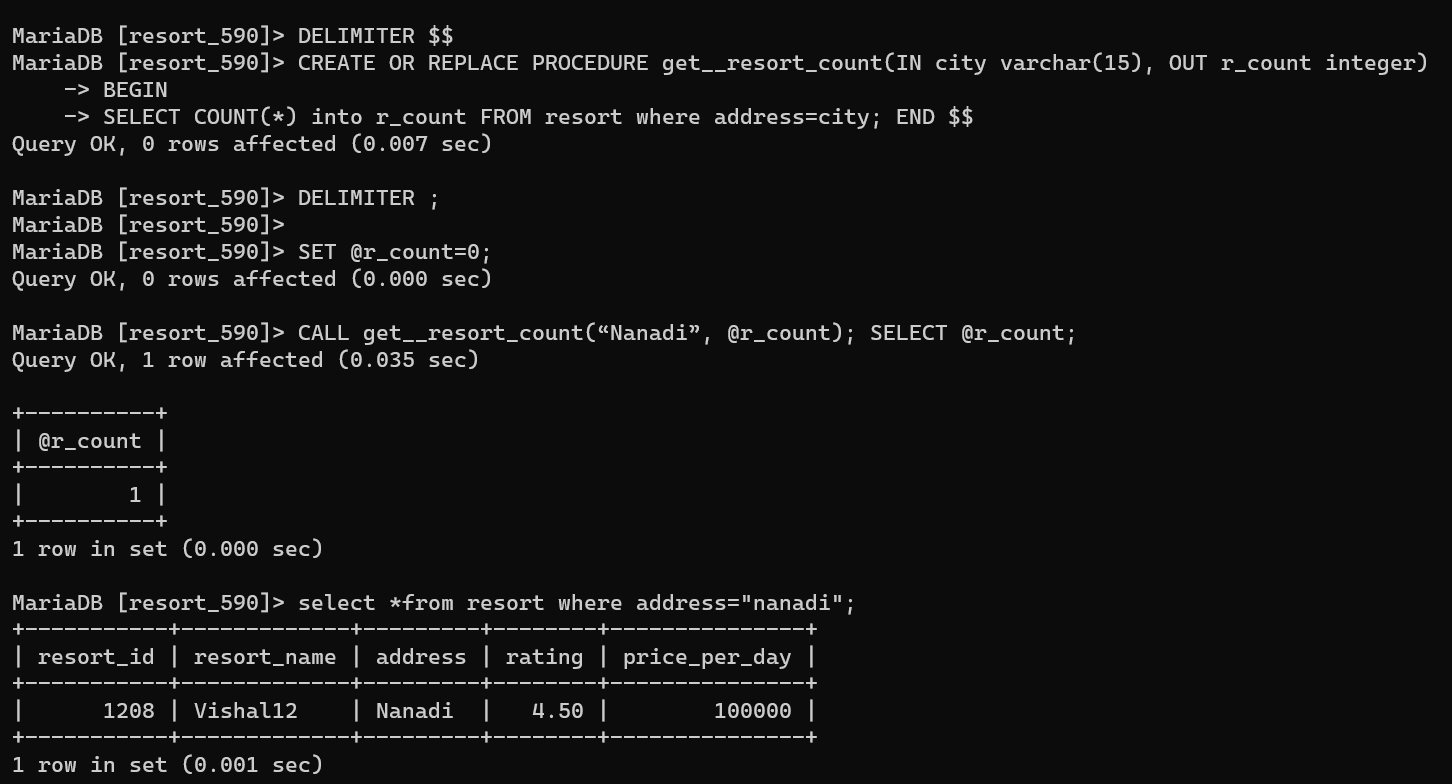
BEGIN

SELECT COUNT(\*) into r\_count FROM resort where address=city; END $$

DELIMITER ;

SET @r\_count=0;

CALL get\_\_resort\_count(“Nanadi”, @r\_count); SELECT @r\_count;



## TRIGGERS

1. display error message when a new entry is added to a resort which is already booked.

DELIMITER $$

CREATE OR REPLACE TRIGGER valid\_reservation\_on\_insert BEFORE INSERT

ON reservation FOR EACH ROW BEGIN

DECLARE error\_msg VARCHAR(255);

SET error\_msg = ("Resort is already booked!");

*-- if checkin for new entry is before checkout for that resort IF NEW.checkin < (select checkout from reservation where*

resort\_id=NEW.resort\_id) THEN SIGNAL SQLSTATE='45000'

SET MESSAGE\_TEXT = error\_msg; END IF;

END $$

CREATE OR REPLACE TRIGGER valid\_reservation\_on\_update BEFORE UPDATE

ON reservation FOR EACH ROW BEGIN

DECLARE error\_msg VARCHAR(255);

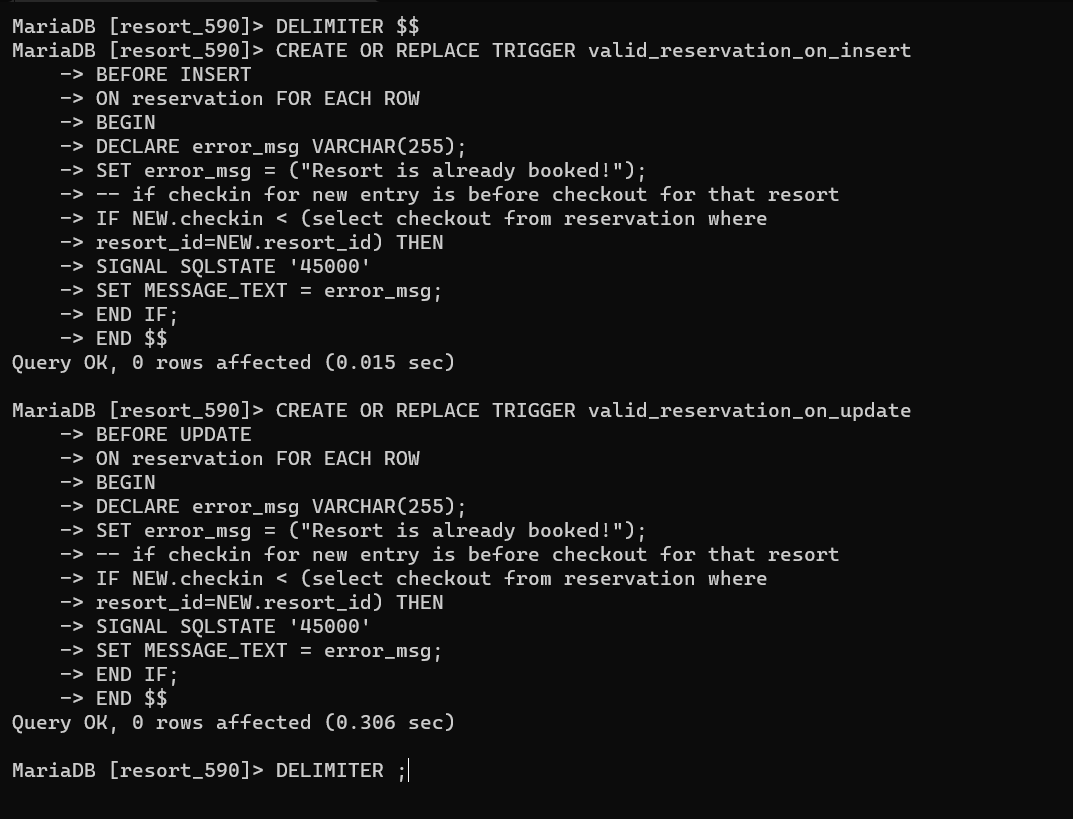
SET error\_msg = ("Resort is already booked!");

*-- if checkin for new entry is before checkout for that resort IF NEW.checkin < (select checkout from reservation where*

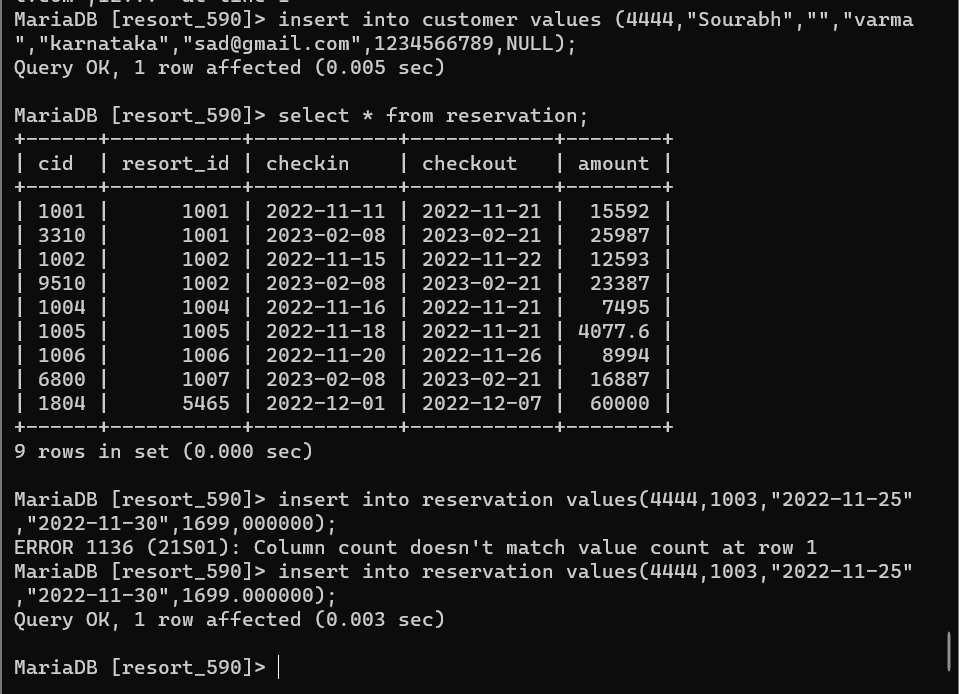
resort\_id=NEW.resort\_id) THEN SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = error\_msg; END IF;

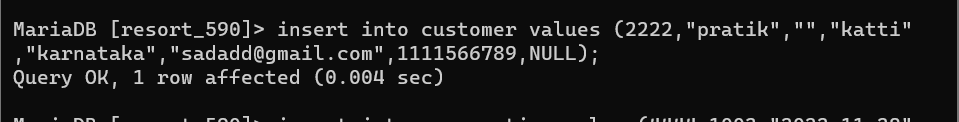
END $$ DELIMITER ;

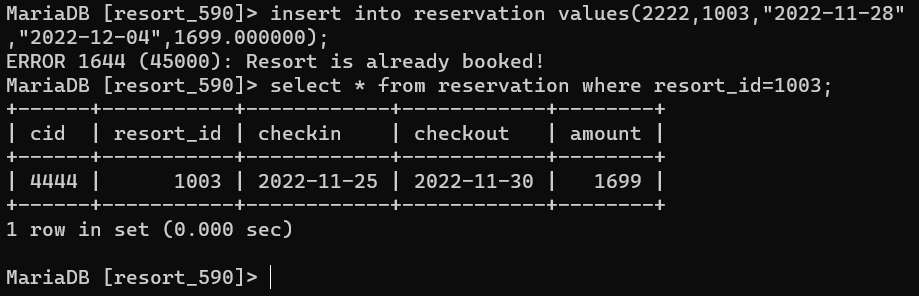


Example

Successful insertion

Case when error is triggered





## CURSORS

Cursor :

To Create backup table for food item

CREATE TABLE `Curbackup\_foodItem` (

  `food\_id` int(11) DEFAULT NULL,

  `food\_name` varchar(30) DEFAULT NULL,

  `price` int(11) DEFAULT NULL

);

DELIMITER $$

CREATE PROCEDURE curs()

BEGIN

DECLARE done INT DEFAULT 0;

DECLARE foodid , price int(10);

DECLARE foodname VARCHAR(20);

DECLARE cur CURSOR FOR SELECT \* FROM food\_item;

DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1;

OPEN cur;

label: LOOP

FETCH cur INTO foodid ,  foodname , price;

INSERT INTO Curbackup\_foodItem VALUES(foodid ,  foodname , price);

IF done = 1 THEN LEAVE label;

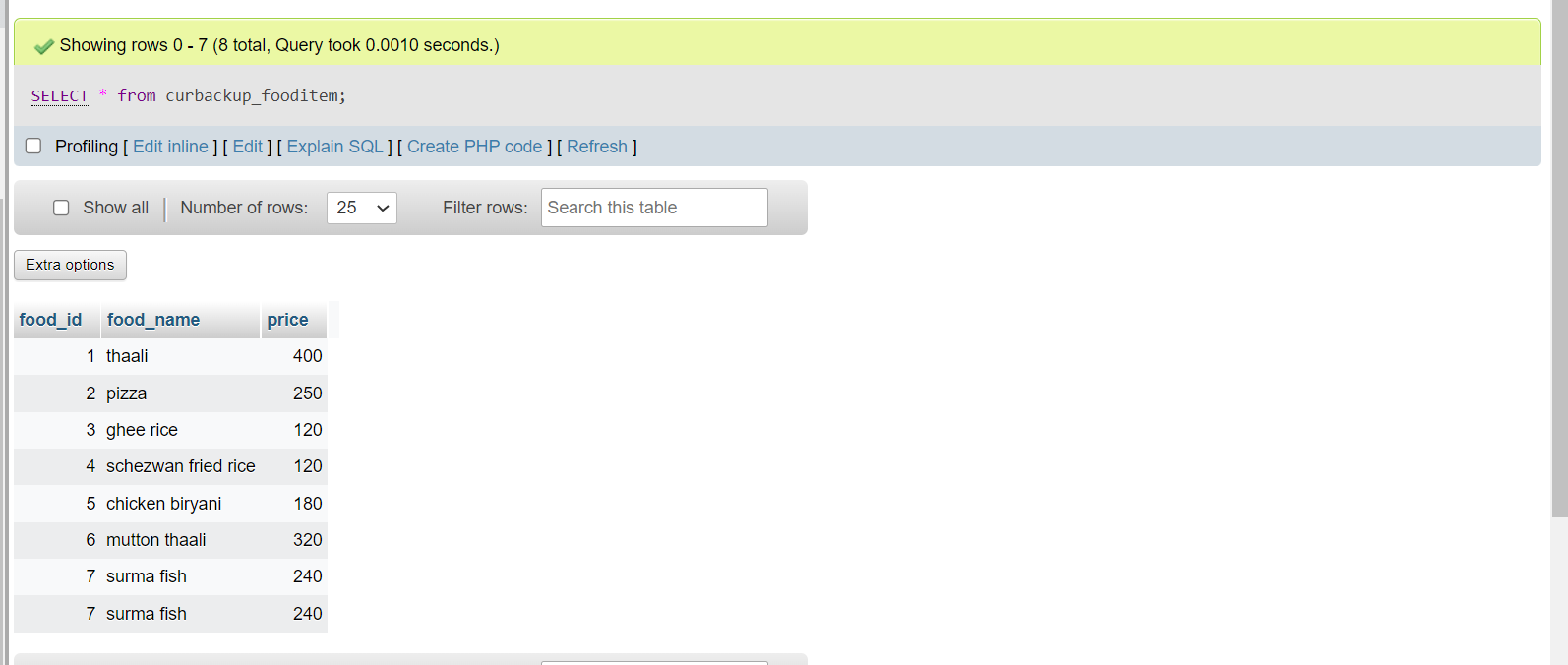
END IF;

END LOOP;

CLOSE cur;

END;$$

DELIMITER ;



## Developing a Frontend

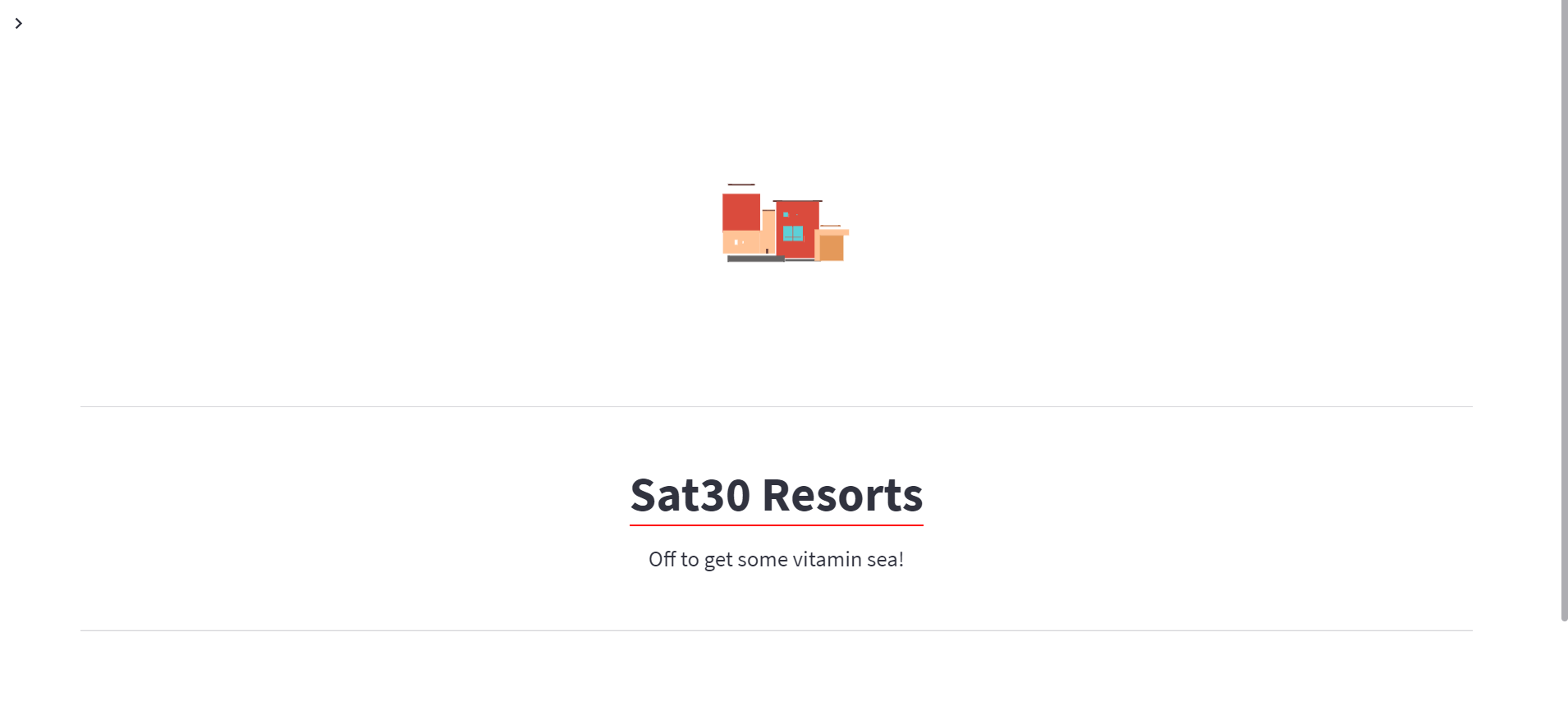
## The frontend should support

## 1. Addition, Modiﬁcation and Deletion of records from any chosen table – Done✅

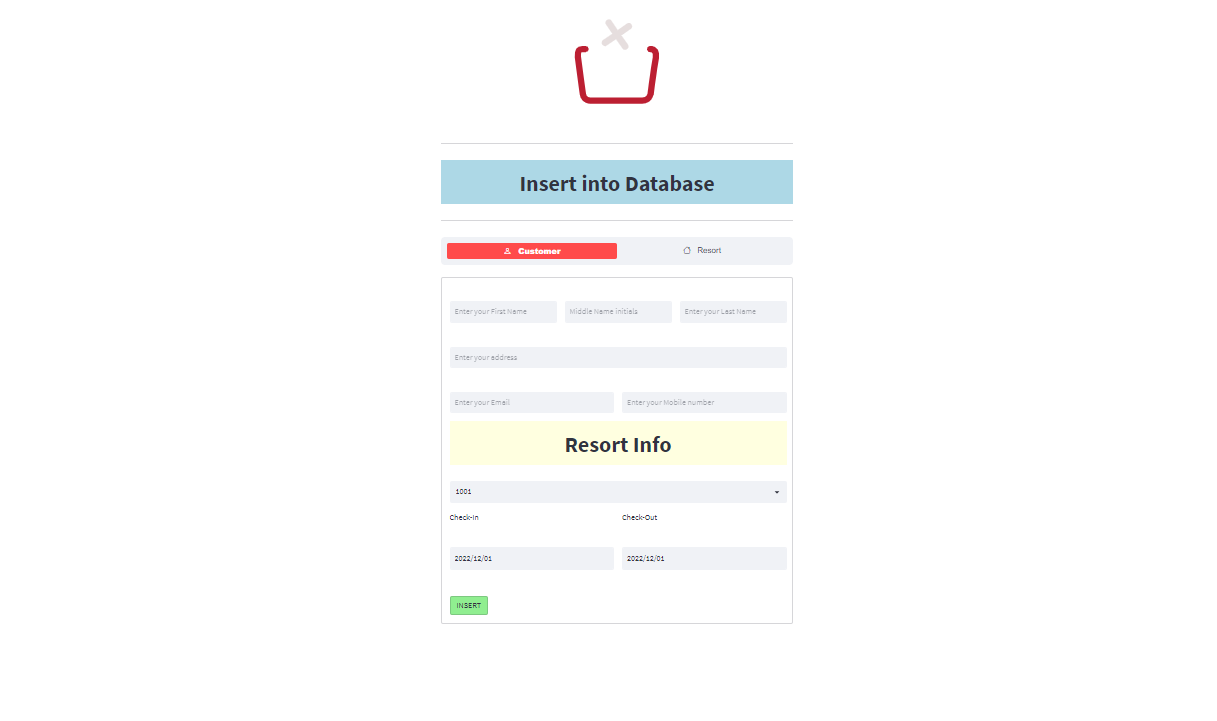
## 2. There should be a window to accept and run any SQL statement and display the result – Done✅

## Frontend made using Streamlit

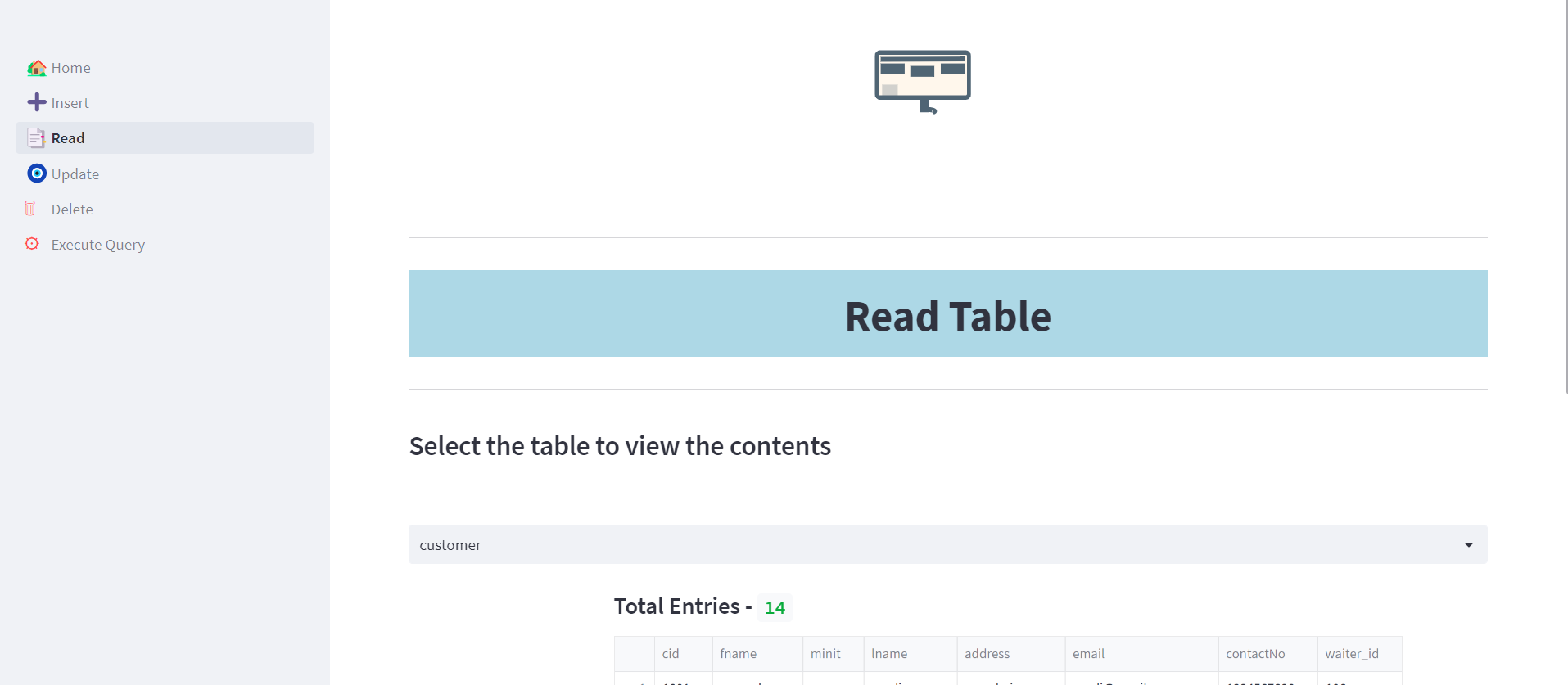
### Frontend Visuals



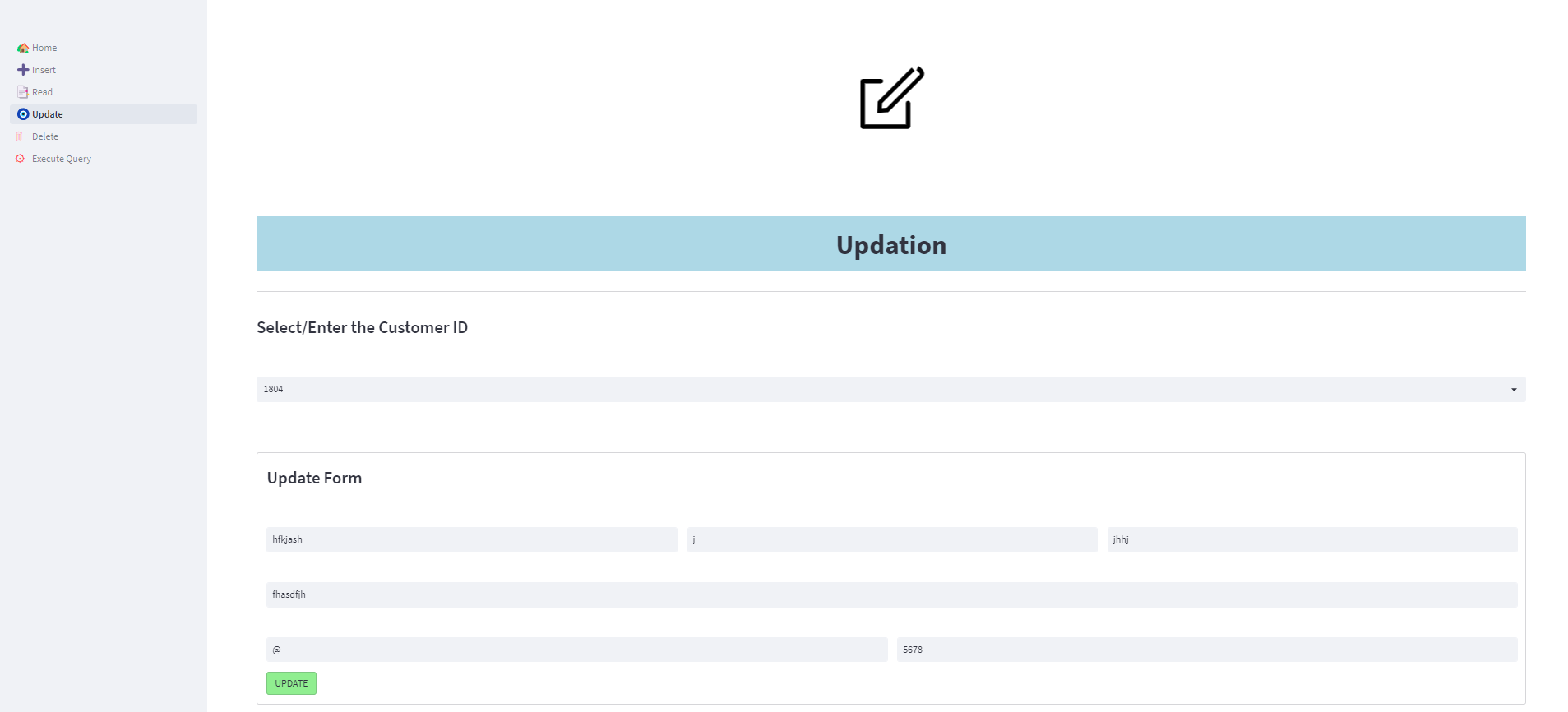
INSERT - OPERATION



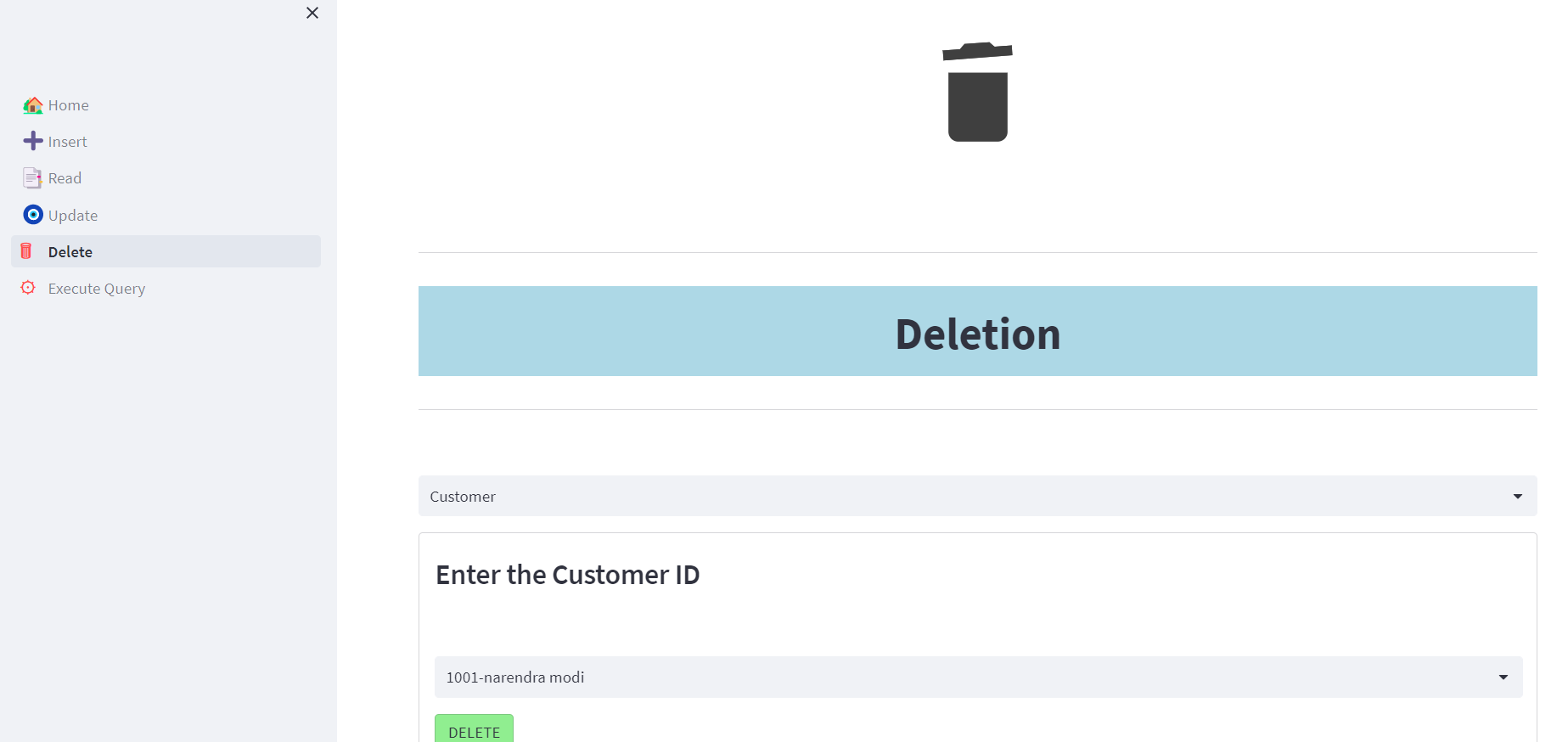
### READ-OPERATION



UPDATE-OPERATION



### DELETE-OPERATION



QUERY-EXECUTION

